



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/071,895	02/08/2002	Philip R. Laws	TD-164	8706

29106 7590 10/01/2004

ROBERT GROOVER III

Groover & Holmes

Box 802889

DALLAS, TX 75380-2889

EXAMINER

JANKUS, ALMIS R

ART UNIT	PAPER NUMBER
----------	--------------

2671

DATE MAILED: 10/01/2004

9

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

10/071,895

Applicant(s)

LAWS ET AL.

Examiner

Almis R Jankus

Art Unit

2671

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☐ Responsive to communication(s) filed on \_\_\_\_.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-43 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-43 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_.

### DETAILED ACTION

1. Applicants' amendment of 6/28/04 has been fully considered in preparing this office action.

2. Applicant's arguments with respect to claims 1-2 have been considered but are moot in view of the new ground(s) of rejection.

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 2, 7, 12, 13, 18, 31, 36, 37 and 43 are rejected under 35 U.S.C. 102(b) as being anticipated by Pineda.

With respect to claim 1, Pineda teaches the claimed method at pages 17-20. In particular, the claimed limitation for calculating edge functions for a patch of pixels, comprising the actions of computing edge function values for at least one interior point within said patch, is taught at page 17 second column at figure 1, and with "Figure 1

Art Unit: 2671

shows how it is possible to define a triangle by the union of three edges which are specified by edge functions. It is possible to define more complex polygons by using boolean combinations of more than three edges.”, and with “With this formalism, it is possible to compute at each pixel center on the plane an n-tuple: (R, G, B, Z, E1...En), where R, G, B and Z components form the fill value, and E1...En are the values of the edge functions which are used to determine whether the pixel is interior or exterior to the polygon. Given the value of this n-tuple at a single pixel position, the n-tuple of adjacent pixels can be computed by simple linear interpolators that require one addition per component per iteration”;

computing edge function values for multiple other points within said patch, at the same paragraph as above, at “Given the value of this n-tuple at a single pixel position, the n-tuple of adjacent pixels can be computed by simple linear interpolators that require one addition per component per iteration”;

using an arithmetic combination of said edge function values for said interior point, together with previously computed values of said edge functions for points on the border of said patch, at page 20 figure 7, at page 18 first column with “We define the edge function  $E(x,y)$  as:

$$E(x,y) = (x - X) dY - (y - Y) dX.$$

This function has the useful property that its value is related to the position of the point (x,y) relative to the edge defined by the points (X,Y) and (X+dX, Y+dY):

$E(x,y) > 0$  if (x,y) is to the "right" side

$E(x,y) = 0$  if (x,y) is exactly on the line

$E(x,y) < 0$  if  $(x,y)$  is to the "l e f t " side",

and at column 18 second column with "Consider a convex polygon defined by the vertices  $(X_i, Y_i)$   $0 < i \leq N$ . For the convenience of notation, take  $(X_0, Y_0) = (X_N, Y_N)$ , and consider the  $i$ 'th edge as the edge between the  $i$ 'th and the  $[i-1]$  vertex. The initial values of the edge function interpolators at a starting point  $(X_s, Y_s)$  would then be:

$$dX_i = X_i - X_{[i-1]}$$

$$dY_i = Y_i - Y_{[i-1]}$$

$$E_i(X_s, Y_s) = (X_s - X_i) dY_i - (Y_s - Y_i) dX_i$$

$$\text{for } 0 < i \leq N$$

The edge functions may then be computed incrementally for a unit step in the X or Y direction:

$$E_i(x+1, y) = E_i(x, y) + dY_i,$$

$$E_i(x-1, y) = E_i(x, y) - dY_i,$$

$$E_i(x, y+1) = E_i(x, y) - dX_i,$$

$$E_i(x, y-1) = E_i(x, y) + dX_i.$$

If we use a tie breaker rule that considers a point on an edge as interior to the edge, then a point is interior to the convex polygon if:

$$E_i \geq 0 \text{ for all } i : 0 < i \leq N.$$

Given the initialized edge interpolators, the interpolation coefficients, the tie breaker rule, and the Boolean function for combining the edges, we still need to traverse the area of the triangle in order to paint it.

Art Unit: 2671

The polygon can be traversed by any algorithm that is guaranteed to cover all of the pixels”;

together with a reduced set of offset vectors, at page 18 first column with “This function is convenient for rasterization algorithms, since it can be computed incrementally by simple addition:

$$E(x+1, y) = E(x, y) + dY$$

$$E(x, y+1) = E(x, y) - dX$$

The edge function is related to the error value or “draw control variable” (DCV) in Bresenham line drawing algorithms [1,2]. The difference is that Bresenham line drawing algorithms maintain the DCV value only for pixels within 1/2 pixel of the line, while  $E(x, y)$  is defined for all pixels on the plane. In addition, the value of the DCV at a given point differs from  $E(x, y)$  by a constant offset. In any case, the reason that both algorithms work is fundamentally the same”, and at page 18 second column with “The edge functions may then be computed incrementally for a unit step in the X or Y direction:

$$E_i(x+1, y) = E_i(x, y) + dY_i,$$

$$E_i(x-1, y) = E_i(x, y) - dY_i,$$

$$E_i(x, y+1) = E_i(x, y) - dX_i,$$

$$E_i(x, y-1) = E_i(x, y) + dX_i,$$

as can be seen, the above equations define vectors with offsets, which are collinear with the primitive edges.

With respect to claim 2 Pineda teaches the claimed computing the value of an edge function at an interior reference point within said patch, at page 17 second column at figure 1, and with "Figure 1 shows how it is possible to define a triangle by the union of three edges which are specified by edge functions. It is possible to define more complex polygons by using boolean combinations of more than three edges.", and with "With this formalism, it is possible to compute at each pixel center on the plane an n-tuple:  $(R, G, B, Z, E_1 \dots E_n)$ , where R, G, B and Z components form the fill value, and  $E_1 \dots E_n$  are the values of the edge functions which are used to determine whether the pixel is interior or exterior to the polygon. Given the value of this n-tuple at a single pixel position, the n-tuple of adjacent pixels can be computed by simple linear interpolators that require one addition per component per iteration.", and at page 19 figure 5, "starting point";

and at a boundary reference point on the edge of said patch, at page 20 figure 7, at page 18 column 1 with "Consider, as shown in figure 2, a vector defined by two points:  $(X, Y)$  and  $(X+dX, Y+dY)$ , and the line that passes through both points. This vector and line can be used to divide the two dimensional space into three regions: all points to the "left" of, to the "right" of, and exactly on the line", and at page 19 at figure 5, and with "A smarter algorithm is shown in figure 5. It proceeds down from the starting point, working its way outward from a center line. The advantage of this algorithm over the simpler algorithm is that it never has to search for an edge, then double back. The tradeoff is that the interpolator state for the center line must be saved while traversing

Art Unit: 2671

the outer points, since the interpolators must be restarted back at the center line. Notice that at the bottom, the "center" line shifts over if it ends up exterior to the triangle."

where it is understood that the teaching of "it never has to search for an edge, then double back" refers to not having to "double back", i.e., the edge must still be searched for corresponding to the claimed boundary reference point, and at column 18 second column with "Consider a convex polygon defined by the vertices  $(X_i, Y_i)$   $0 < i \leq N$ . For the convenience of notation, take  $(X_0, Y_0) = (X_N, Y_N)$ , and consider the  $i$ 'th edge as the edge between the  $i$ 'th and the  $[i-1]$  vertex. The initial values of the edge function interpolators at a starting point  $(X_s, Y_s)$  would then be:

$$dX_i = X_i - X_{[i-1]}$$

$$dY_i = Y_i - Y_{[i-1]}$$

$$E_i(X_s, Y_s) = (X_s - X_i) dY_i - (Y_s - Y_i) dX_i$$

$$\text{for } 0 < i \leq N$$

The edge functions may then be computed incrementally for a unit step in the X or Y direction:

$$E_i(x+1, y) = E_i(x, y) + dY_i,$$

$$E_i(x-1, y) = E_i(x, y) - dY_i,$$

$$E_i(x, y+1) = E_i(x, y) - dX_i,$$

$$E_i(x, y-1) = E_i(x, y) + dX_i.$$

If we use a tie breaker rule that considers a point on an edge as interior to the edge, then a point is interior to the convex polygon if:

$$E_i \geq 0 \text{ for all } i : 0 < i \leq N.$$



Art Unit: 2671

Given the initialized edge interpolators, the interpolation coefficients, the tie breaker rule, and the Boolean function for combining the edges, we still need to traverse the area of the triangle in order to paint it.

The polygon can be traversed by any algorithm that is guaranteed to cover all of the pixels”;

assessing the value of said edge function at multiple other points within said patch, by comparing the value of said edge function at a respective reference point, which may be said interior reference point or said boundary reference point or a previously computed reference point, at page 18 first column with “We define the edge function  $E(x,y)$  as:

$$E(x,y) = (x - X) dY - (y - Y) dX.$$

This function has the useful property that its value is related to the position of the point  $(x,y)$  relative to the edge defined by the points  $(X,Y)$  and  $(X+dX, Y+dY)$ :

$E(x,y) > 0$  if  $(x,y)$  is to the "right" side

$E(x,y) = 0$  if  $(x,y)$  is exactly on the line

$E(x,y) < 0$  if  $(x,y)$  is to the "l e f t " side”,

and at page 19 second column with “The polygon can be traversed by any algorithm that is guaranteed to cover all of the pixels”;

with the delta value of said edge function for a respective one of a reduced set of offset vectors, wherein said reduced set of offset vectors does not include vectors which are complements or shifts of each other, at page 18 first column which teaches the delta value offsets which are used to rasterize by simple addition, with “This function is

Art Unit: 2671

convenient for rasterization algorithms, since it can be computed incrementally by simple addition:

$$E(x+1, y) = E(x, y) + dV$$

$$E(x, y+1) = E(x, y) - dX$$

The edge function is related to the error value or "draw control variable" (DCV) in Bresenham line drawing algorithms [1,2]. The difference is that Bresenham line drawing algorithms maintain the DCV value only for pixels within 1/2 pixel of the line, while  $E(x, y)$  is defined for all pixels on the plane. In addition, the value of the DCV at a given point differs from  $E(x, y)$  by a constant offset. In any case, the reason that both algorithms work is fundamentally the same", and at page 18 second column with "The edge functions may then be computed incrementally for a unit step in the X or Y direction:

$$E_i(x+1, y) = E_i(x, y) + dY_i,$$

$$E_i(x-1, y) = E_i(x, y) - dY_i,$$

$$E_i(x, y+1) = E_i(x, y) - dX_i,$$

$$E_i(x, y-1) = E_i(x, y) + dX_i,$$

as can be seen, the above equations define vectors with offsets, which are collinear with the primitive edges. The vectors are neither shifts nor complements of each other, rather, they are vectors which incorporate delta offset complements, for example,  $+dY_i$ ,  $-dY_i$ . It is noted that this last limitation of "reduced set of offset vectors, wherein said reduced set of offset vectors does not include vectors which are complements or shifts of each other" is not described in the specification in such a way as to enable one

skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. However, the examiner is including a definition, as provided above, for what might reasonably be construed as a reduced set of offset vectors which do not include vectors which are complements or shifts of each other.

With respect to claim 13, Pineda teaches the claimed step for calculating a half-plane membership function value for at least one reference point pixel, at the boundary of a first patch of pixels, at page 20 figure 7, at page 18 column 1 with "Consider, as shown in figure 2, a vector defined by two points: (X,Y) and (X+dX,Y+dY), and the line that passes through both points. This vector and line can be used to divide the two dimensional space into three regions: all points to the "left" of, to the "right" of, and exactly on the line", at page 18 first column with "We define the edge function E(x,y) as:

$$E(x,y) = (x - X) dY - (y - Y) dX.$$

This function has the useful property that its value is related to the position of the point (x,y) relative to the edge defined by the points (X,Y) and (X+dX, Y+dY):

$$E(x,y) > 0 \text{ if } (x,y) \text{ is to the "right" side}$$

$$E(x,y) = 0 \text{ if } (x,y) \text{ is exactly on the line}$$

$$E(x,y) < 0 \text{ if } (x,y) \text{ is to the "left" side};$$

a step for calculating values of said half-plane membership function for at least some pixels in said patch, at page 17 second column at figure 1, and with "Figure 1 shows how it is possible to define a triangle by the union of three edges which are specified by edge functions. It is possible to define more complex polygons by using

boolean combinations of more than three edges.”, and with “With this formalism, it is possible to compute at each pixel center on the plane an n-tuple: (R, G, B, Z, E1...En), where R, G, B and Z components form the fill value, and E1...En are the values of the edge functions which are used to determine whether the pixel is interior or exterior to the polygon. Given the value of this n-tuple at a single pixel position, the n-tuple of adjacent pixels can be computed by simple linear interpolators that require one addition per component per iteration”;

and a step for using an inheritance relation to carry forward the half-plane membership function values of said reference points for a successive patch of pixels, at pages 19-20 with “Since the edge function is linear, it is possible to compute the value of the edge function for a pixel an arbitrary distance L away from a given point (x,y) :

$$E(x+L, y) = E(x) + L \, dy$$

This property allows a group of interpolators, each responsible for a pixel within a block of contiguous pixels, to simultaneously compute the edge function of an adjacent block in a single cycle. If the blocks were L pixels wide, then there would be L interpolators. In order to compute the edge function of the block L pixels away in the +x direction, each interpolator would increment by (L dx). Since color and Z components are linear as well, they may also be computed in parallel”;

wherein the computational load is minimized, at page 20 with “Graphics frame buffers are usually organized to provide simultaneous access to a block of adjacent pixels [5]. The block is usually called a word, and the pixels within the block are called interleaves. If a group of interpolators are dedicated to each interleave, then the RGBZ

Art Unit: 2671

value and whether the pixel should be drawn can be computed in parallel for an entire word. If the interpolator cycle time is at least as fast as the memory cycle time (which is the case with current, gate array and DRAM technology), then shaded triangles can be rendered at the memory cycle time".

With respect to claim 31, Pineda teaches the claimed device for calculating the half-plane membership function value for pixels at the boundaries of a first patch of pixels for use as reference points, at page 20 figure 7, at page 18 column 1 with "Consider, as shown in figure 2, a vector defined by two points: (X,Y) and (X+dX,Y+dY), and the line that passes through both points. This vector and line can be used to divide the two dimensional space into three regions: all points to the "left" of, to the "right" of, and exactly on the line", at page 18 first column with "We define the edge function E(x,y) as:

$$E(x,y) = (x - X) dY - (y - Y) dX.$$

This function has the useful property that its value is related to the position of the point (x,y) relative to the edge defined by the points (X,Y) and (X+dX, Y+dY):

$E(x,y) > 0$  if (x,y) is to the "right" side

$E(x,y) = 0$  if (x,y) is exactly on the line

$E(x,y) < 0$  if (x,y) is to the "l e f t " side";

and rapidly computing half-plane membership function values for successive patches of pixels, by using an inheritance relation to carry forward the half-plane membership function values of said reference points for ones of said successive patch

of pixels, at pages 19-20 with "Since the edge function is linear, it is possible to compute the value of the edge function for a pixel an arbitrary distance L away from a given point (x,y) :

$$E(x+L, y) = E(x) + L \, dy$$

This property allows a group of interpolators, each responsible for a pixel within a block of contiguous pixels, to simultaneously compute the edge function of an adjacent block in a single cycle. If the blocks were L pixels wide, then there would be L interpolators. In order to compute the edge function of the block L pixels away in the +x direction, each interpolator would increment by (L dx). Since color and Z components are linear as well, they may also be computed in parallel";

wherein the computational load for each successive patch is minimized, at page 20 with "Graphics frame buffers are usually organized to provide simultaneous access to a block of adjacent pixels [5]. The block is usually called a word, and the pixels within the block are called interleaves. If a group of interpolators are dedicated to each interleave, then the RGBZ value and whether the pixel should be drawn can be computed in parallel for an entire word, If the interpolator cycle time is at least as fast as the memory cycle time (which is the case with current, gate array and DRAM technology), then shaded triangles can be rendered at the memory cycle time", and system with host, at page 18 with "this same property of E(x,y) is used by the "PIXEL-PLANES" [3] graphics system, where this function is computed in parallel for all pixels in the frame buffer by a multiplier tree".

With respect to claim 37, Pineda teaches the claimed computer system comprising a host processor and a 3D graphics rasterizer, at the abstract, and with the "PIXEL-PLANES" system; comprising means for generating reference point values by evaluating the half-plane membership function value at one or more reference points in a tile of pixels, at page 17 right column, at figures 1, 2, 3, 6, and 7; means for referencing a pixel within said tile against one of said reference points, at page 17 right column and at page 18; means for stating the position of said pixel as an offset from said reference point, at pages 17-20 and at figures 2-7; means for shading said pixel according to the reference point value, at page 17 right column (RGBZ); and means for determining whether said pixel is inside or outside of an edge by comparing the reference point value with the offset to said pixel, at page 18 as the "tie breaker rule".

Claims 7, 12, 18, 36 and 43 depend from claims 1, 2, 13, 31 and 37 and further require a pixel within said patch to be referenced against at least one of said edge function values (reference point values: claims 12, 18, 36, 43). Pineda teaches referencing against at least one of the edge function values at page 18 section 2 and referencing against at least one reference point value at sections 3 and 4 and with the teaching of incremental polygon traversal. With incremental traversal (walking), a current position and/or value of a pixel is dependent on the position and/or value of the previously visited pixel. Therefore, and incremented pixel is referenced against a previous pixel's (reference point's) value (position, color, depth, etc.).

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 3-6, 8-11, 14-17, 32-35, 38-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pineda as applied to claims 1, 2, 13, 31 and 37 above, and further in view of Apgar et al.

Claims 3, 8, 14, 32 and 39 depend from claims 1, 2, 13, 31 and 37 respectively, and further require the patch to comprise 64 pixels (note that the term "patch" is changed in claim 32 to the phrase "patch of pixels", and claim 39 uses the term "tile"; all of these terminology variations are taken to mean the same feature). Further, claims 4,



9, 15, 33 and 40 depend from claims 1, 2, 13, 31 and 37 respectively, and require the patch to be an 8x8 tile.

Apgar et al. teach a computer system which also uses bounding functions to determine which pixels are interior to a polygon. For triangles and lines, the bounding functions are the linear equations defining the sides or ends of the primitive. It uses a "Foot Print Engine" which is a 4x4 array of "Toe" ASICs arranged so each Toe handles one pixel per memory block. Figure 4 shows the 4x4 pixel blocks covering a triangle, and a footprint position with enabled toes. Apgar et al. further teach using the claimed patch as an 8x8 tile, at page 261 with the teaching of "using an 8x8 array of Toes" ("Toes" correspond to pixels; array corresponds to tile). The limitation of a 64 pixel patch is satisfied with a teaching of an 8x8 tile because 8 times 8 equals 64. It would have been obvious to one of ordinary skill in the art to use the teachings of Apgar et al. with the teachings of Pineda because Pineda teaches parallel computation for blocks of pixels (figure 7); and Apgar et al. perform this parallel computation for blocks of pixels; additionally, Apgar et al. provide the motivation for using the 8x8 tile at page 261 with "Simulations have shown that the Foot Print processor architecture can be scaled to approximately 3 times the present performance by using an 8x8 array of Toes. Four times the memory bandwidth would be needed to support this, but such bandwidth seems possible. Toe arrays larger than 8x8 would suffer from poor efficiency and could not be cost-justified".

Claims 5, 10, 16, 34 and 41 depend from claims 1, 2, 13, 31 and 37 respectively, and further require only one interior point (interior reference point) edge function value and the edge function value for one new point on said border (boundary reference point) are computed for a successive patch. Apgar et al. teach this at page 259 at figure 4, at figure 6 which shows the interior point as (1,1) containing the coefficient "C", and with "Under the control of the Address Engine, the Foot Print Engine begins a primitive by fetching the initial values and coefficients for all the rendering equations from the transfer RAM. These initial values were computed for the pixel at position (1,1) in the first block of the primitive. Each Toe adjusts the values (by adding or subtracting the appropriate coefficients, as shown in Figure 6) to obtain the correct values for its position in the array", and the new point on the border as determined by the Address Engine, with the teaching of "The Address Engine begins a primitive by fetching the relevant data from the transfer RAM in the Setup Engine. The Address Engine evaluates the bounding function(s) of the primitive, sequentially determining the virtual address of each 4x4 block that the primitive hits (See Figure 4.)" and with "After the first block is done, the Address Engine walks the Foot Print Engine to the other blocks in horizontal or vertical steps, at each step moving 4 pixels in either X or Y. For a horizontal step, each Toe updates each function by 4A, while for vertical steps the functions are updated by 4B. The functions are evaluated by finite differences taking advantage of the coherence of the objects being rendered". It would have been obvious to one of ordinary skill in the art to use the teachings of Apgar et al. with the teachings of Pineda because Pineda teaches parallel computation for blocks of pixels (figure 7);

and Apgar et al. perform this parallel computation for blocks of pixels using the bounding functions of Pineda, resulting in fast parallel computation of primitives where functions are evaluated by finite differences taking advantage of the coherence of the objects being rendered.

Claims 6, 11, 17, 35 and 42 depend from claims 1, 2, 13, 31 and 37 respectively, and further require said patch to have one interior point (interior reference point) edge function value and edge function values for four points on said border (boundary reference points). Apgar et al. teaches this at figure 6 and with the teaching of the Address Engine walking the Foot Print Engine to other blocks. It would have been obvious to one of ordinary skill in the art to use the teachings of Apgar et al. with the teachings of Pineda because Pineda teaches parallel computation for blocks of pixels (figure 7); and Apgar et al. perform this parallel computation for blocks of pixels using the bounding functions of Pineda, resulting in fast parallel computation of primitives where functions are evaluated by finite differences taking advantage of the coherence of the objects being rendered.

Claim 38 depends from claim 37, and further requires said reference points to be the top left pixel, the top right pixel, the bottom left pixel, the bottom right pixel, and the middle pixel of said tile. Apgar et al. teaches this at figure 6 and with the teaching of the Address Engine walking the Foot Print Engine to other blocks. The corner pixels define the new location of the foot print, and the middle pixel contains the initial values updated

by either 4A or 4B depending on whether the walk is horizontal or vertical respectively. It would have been obvious to one of ordinary skill in the art to use the teachings of Apgar et al. with the teachings of Pineda because Pineda teaches parallel computation for blocks of pixels (figure 7); and Apgar et al. perform this parallel computation for blocks of pixels using the bounding functions of Pineda, resulting in fast parallel computation of primitives where functions are evaluated by finite differences taking advantage of the coherence of the objects being rendered.

8. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

9. Claims 2-12, 19-24 and 25-30 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

With respect to claim 2, the limitation of a "reduced set of offset vectors, wherein said reduced set of offset vectors does not include vectors which are complements or shifts of each other" is not described in the specification in accordance with the requirements of 35 U.S.C. 112, first paragraph. A reduced set of offset vectors

is mentioned in the specification at page 14 lines 17-18 as an item to be included in a calculation, however, nowhere in the specification is there any description of this reduced set of offset vectors. At page 14 line 28 to page 15 line 2, a reduced set of offset vectors wherein said reduced set of offset vectors does not include vectors which are complements or shifts of each other is mentioned as an item to be included in another calculation, but again nowhere in the specification is there any description of this reduced set of offset vectors.

With respect to claims 19 and 25, the limitation of "symmetry relations" is not described in the specification in accordance with the requirements of 35 U.S.C. 112, first paragraph. At page 5 lines 13-21 (the phrase is found at line 17) the instant specification indicates that the present application describes a set of techniques, including symmetry relations, for rapidly computing a function for patches of pixels. However, symmetry relations are nowhere mentioned again in the specification, therefore they are not described.


Claims 8-12, 20-24 and 26-30 depend from claims 2, 19 and 25 respectively, and are rejected based on lack of enablement of their respective parent claims.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Almis R Jankus whose telephone number is 703-305-9795. The examiner can normally be reached on M-F, 8-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mark Zimmerman can be reached on 703-305-9798. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJ



ALMIS R. JANKUS  
PRIMARY EXAMINER